**Day-1 Assignment**

**Class and object JAVA**

1.1 **List All Even Numbers.**

**package** org.software.interfacexample;

**import** java.util.Scanner;

**public** **class** ListEvenNumbers {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter Number : ");

**int** num = sc.nextInt();

System.***out***.println("Even Numbers Less Than or Equal " + num + " are:");

**for** (**int** i = 1; i <= num; i++) {

**if** (i % 2 == 0)

System.***out***.print(i + " ");

}

}

}

**1.2 Rectangle class.**

**package** org.mphasis.rectangle;

**public** **class** RectAngle {

**double** length, breadth;

**double** area;

**public** RectAngle() {

**this**.length = 0;

**this**.breadth = 0;

}

**public** RectAngle(**double** length, **double** breadth) {

**this**.length = length;

**this**.breadth = breadth;

}

**void** calculateArea() {

area=length\*breadth;

}

**void** toDisplay() {

System.***out***.println("---------------RECTANGLE-----------------");

System.***out***.println("Length of Rectangle : "+length);

System.***out***.println("breadth of Rectangle : "+breadth);

System.***out***.println("Area of Rectangle : "+area);

}

**public** **double** getLength() {

**return** length;

}

**public** **void** setLength(**double** length) {

**this**.length = length;

}

**public** **double** getBreadth() {

**return** breadth;

}

**public** **void** setBreadth(**double** breadth) {

**this**.breadth = breadth;

}

}

**TestRectangle.**

**package** org.mphasis.rectangle;

**import** java.util.Scanner;

**import** org.mphasis.onefive.RectanglePerimeter;

**public** **class** TestRectangle {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

RectanglePerimeter rp=**new** RectanglePerimeter();

System.***out***.println("Enter length");

**double** l=sc.nextDouble();

System.***out***.println("Enter Width");

**double** b=sc.nextDouble();

RectAngle ra=**new** RectAngle(l,b);

ra.calculateArea();

ra.toDisplay();

}

}

**1.3 Books class.**

**package** org.mphasis.firstp;

**public** **class** Books {

String book\_title;

**double** book\_price;

**public** Books() {

}

**public** String getBook\_title() {

**return** book\_title;

}

**public** **void** setBook\_title(String book\_title) {

**this**.book\_title = book\_title;

}

**public** **double** getBook\_price() {

**return** book\_price;

}

**public** **void** setBook\_price(**double** book\_price) {

**this**.book\_price = book\_price;

}

}

**TestBook class.**

package org.mphasis.firstp;

import java.util.Iterator;

import java.util.Scanner;

public class TestBook {

private static Books createBooks() {

Scanner sc=new Scanner(System.in);

Books b=new Books();

System.out.println("Enter book Title");

b.setBook\_title(sc.next());

System.out.println("Enter book Price");

b.setBook\_price(sc.nextDouble());

return b;

}

private static void showBooks(Books[] books) {

System.out.println("BOOK TITLE " +" | "+"BOOK PRICE" );

for (int i = 0; i < books.length; i++) {

System.out.println(books[i].getBook\_title() +" | "+books[i].getBook\_price());

}

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("How May Books You Want to Enter");

int n=sc.nextInt();

Books books[]=new Books[n];

for (int i = 0; i < books.length; i++) {

Books b=createBooks();

books[i]=b;

}

showBooks(books);

}

}

**1.4 RectanglePerimeter class.**

**package** org.mphasis.onefive;

**import** java.util.Scanner;

**public** **class** RectanglePerimeter {

**double** length = 1.0;

**double** width = 1.0;

**public** **double** getLength() {

**return** length;

}

**public** **void** setLength(**double** length) {

**if** (length > 0.0 && length < 20.0)

**this**.length = length;

}

**public** **double** getWidth() {

**return** width;

}

**public** **void** setWidth(**double** width) {

**if** (width > 0.0 && width < 20.0)

**this**.width = width;

}

**public** **void** calculatePerimeterArea() {

**double** perimeter = 2 + (length + width);

System.***out***.println("The Perimeter of a Rectangle : " + perimeter);

**double** area = length \* width;

System.***out***.println("The Area of a Rectangle : " + area);

}

}

**Test class.**

**package** org.mphasis.onefive;

**import** java.util.Scanner;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

RectanglePerimeter rp=**new** RectanglePerimeter();

System.***out***.println("Enter length");

rp.setLength(sc.nextDouble());

System.***out***.println("Enter Width");

rp.setWidth(sc.nextDouble());

**if**(rp.getLength()>1.0 && rp.getWidth()>1.0)

rp.calculatePerimeterArea();

**else**

System.***out***.println("Invalid Data Entry");

}

}

**1.5 TestEmployee Class.**

**package** org.abstraction.com;

**import** java.time.LocalDate;

**import** java.time.format.DateTimeParseException;

**import** java.util.Scanner;

**public** **class** TestEmployee {

**public** **static** **void** main(String[] args) {

**try** (Scanner scanner = **new** Scanner(System.***in***)) {

Employee[] employees = **new** Employee[5];

**for** (**int** i = 0; i < 5; i++) {

System.***out***.println("Enter details for employee " + (i + 1));

System.***out***.print("Enter employee number: ");

**while** (!scanner.hasNextInt()) {

System.***out***.println("Invalid input. Please enter a valid employee number.");

scanner.next();

}

**int** employeeNumber = scanner.nextInt();

scanner.nextLine();

System.***out***.print("Enter employee name: ");

String employeeName = scanner.nextLine();

System.***out***.print("Enter joining date (YYYY-MM-DD): ");

String dateInput = scanner.nextLine();

**try** {

LocalDate joiningDate = LocalDate.*parse*(dateInput);

employees[i] = **new** Employee(employeeNumber, employeeName, joiningDate);

} **catch** (DateTimeParseException e) {

System.***out***.println("Invalid date format. Please enter the details again.");

i--;

}

}

**for** (Employee employee : employees) {

System.***out***.println("\n" + employee);

}

}

}

}

**class** Employee {

**private** **int** employeeNumber;

**private** String employeeName;

**private** LocalDate joiningDate;

**public** Employee(**int** employeeNumber, String employeeName, LocalDate joiningDate) {

**this**.employeeNumber = employeeNumber;

**this**.employeeName = employeeName;

**this**.joiningDate = joiningDate;

}

**public** **int** getEmployeeNumber() {

**return** employeeNumber;

}

**public** String getEmployeeName() {

**return** employeeName;

}

**public** LocalDate getJoiningDate() {

**return** joiningDate;

}

**public** String toString() {

**return** "Employee Number: " + employeeNumber + "\nEmployee Name: " + employeeName + "\nJoining Date: "

+ joiningDate;

}

}

**Problem Statement 2: Encapsulation and Inheritance in Java OOPs**

**Vehicle class.**

**package** org.mphasis.encapsulation;

**import** java.sql.Date;

**class** Vehicle {

String manufacturer;

String model;

**int** year;

**void** displayDetails() {

}

**public** String getManufacturer() {

**return** manufacturer;

}

**public** **void** setManufacturer(String manufacturer) {

**this**.manufacturer = manufacturer;

}

**public** String getModel() {

**return** model;

}

**public** **void** setModel(String model) {

**this**.model = model;

}

**public** **int** getYear() {

**return** year;

}

**public** **void** setYear(**int** year) {

**this**.year = year;

}

}

**Car class.**

**package** org.mphasis.encapsulation;

**class** Car **extends** Vehicle {

**int** seatingCapacity;

@Override

**void** displayDetails() {

System.***out***.println(getManufacturer());

System.***out***.println(getModel());

System.***out***.println(getSeatingCapacity());

System.***out***.println(getYear());

}

**void** accelerate() {

System.***out***.println("Accelerated");

}

**void** brake() {

System.***out***.println("Break Done");

}

**public** **int** getSeatingCapacity() {

**return** seatingCapacity;

}

**public** **void** setSeatingCapacity(**int** seatingCapacity) {

**this**.seatingCapacity = seatingCapacity;

}

}

**MotorCycle Class.**

**package** org.mphasis.encapsulation;

**class** MotorCycle **extends** Vehicle {

**double** engineCapacity;

@Override

**void** displayDetails() {

System.***out***.println(getManufacturer());

System.***out***.println(getModel());

System.***out***.println(getEngineCapacity());

System.***out***.println(getYear());

}

**void** startEngine() {

System.***out***.println("Engine Started Successfully");

}

**void** stopEngine() {

System.***out***.println("Engine Stopped Successfully");

}

**public** **double** getEngineCapacity() {

**return** engineCapacity;

}

**public** **void** setEngineCapacity(**double** engineCapacity) {

**this**.engineCapacity = engineCapacity;

}

}

**Truck Class.**

**package** org.mphasis.encapsulation;

**class** Truck **extends** Vehicle {

**double** cargoCapacity;

@Override

**void** displayDetails() {

System.***out***.println(getManufacturer());

System.***out***.println(getModel());

System.***out***.println(getCargoCapacity());

System.***out***.println(getYear());

}

**void** loadCargo() {

System.***out***.println("Cargo Loaded Successfully");

}

**void** unloadCargo() {

System.***out***.println("Cargo Unloaded Successfully");

}

**public** **double** getCargoCapacity() {

**return** cargoCapacity;

}

**public** **void** setCargoCapacity(**double** cargoCapacity) {

**this**.cargoCapacity = cargoCapacity;

}

}

**Main Class.**

**package** org.mphasis.encapsulation;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

System.***out***.println("--------------------CAR------------------------");

Car c=**new** Car();

c.setSeatingCapacity(4);

c.setModel("Mercedes-Benz C-Class");

c.setManufacturer("Mercedes-Benz");

c.setYear(2020);

c.displayDetails();

c.accelerate();

c.brake();

System.***out***.println("---------------------MOTOR CYCLE-----------------------");

MotorCycle mc=**new** MotorCycle();

mc.setEngineCapacity(2);

mc.setModel("Harley-Davidson Street Glide");

mc.setManufacturer("Harley-Davidson");

mc.setYear(2022);

mc.displayDetails();

mc.startEngine();

mc.stopEngine();

System.***out***.println("----------------------TRUCK----------------------");

Truck tr=**new** Truck();

tr.setCargoCapacity(50);

tr.setModel("Ford F-150");

tr.setManufacturer("Ford");

tr.setYear(2023);

tr.displayDetails();

tr.loadCargo();

tr.unloadCargo();

}

}

**Problem Statement 3: Abstraction in Java OOPs**

**Shape Class.**

**package** org.mphasis.abstraction;

**abstract** **class** Shape {

**abstract** **void** calculateArea();

}

**Circle Class.**

**package** org.mphasis.abstraction;

**public** **class** Circle **extends** Shape {

**float** radius;

**public** Circle(**float** radius) {

**super**();

**this**.radius = radius;

}

@Override

**void** calculateArea() {

**double** area = 3.14 \* radius \* radius;

System.***out***.println("The Area of Triangle is : " + area);

} }

**Rectangle Class.**

**package** org.mphasis.abstraction;

**public** **class** Rectangle **extends** Shape {

**float** length, breadth;

**public** Rectangle(**float** length, **float** breadth) {

**super**();

**this**.length = length;

**this**.breadth = breadth;

}

@Override

**void** calculateArea() {

**double** area = length \* breadth;

System.***out***.println("The Area of Rectangle is : " + area);

}

}

**Triangle Class.**

**package** org.mphasis.abstraction;

**public** **class** Triangle **extends** Shape{

**float** height, base;

**public** Triangle(**float** height, **float** base) {

**super**();

**this**.height = height;

**this**.base = base;

}

@Override

**void** calculateArea() {

**double** area=height\*base/2;

System.***out***.println("The Area of Triangle is : " + area);

}

}

**Main Class.**

**package** org.mphasis.abstraction;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

System.***out***.println("--------------------CIRCLE------------------------");

Circle c = **new** Circle(10);

c.calculateArea();

System.***out***.println("--------------------RECTANGLE------------------------");

Rectangle rc = **new** Rectangle(10, 20);

rc.calculateArea();

System.***out***.println("--------------------TRIANGLE------------------------");

Triangle tr = **new** Triangle(13, 15);

tr.calculateArea();

}

}

**Problem Statement 4: Implement the Static Classes and Methods in Java**

**Employees Class.**

**package** mphasis.com;

**public** **class** Employees {

**private** String name;

**private** **int** points;

**private** **static** **int** *employeeCount* = 0;

**public** Employees(String name, **int** points) {

**this**.name = name;

**this**.points = points;

*employeeCount*++;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getPoints() {

**return** points;

}

**public** **void** setPoints(**int** points) {

**this**.points = points;

}

**public** **static** **int** getEmployeeCount() {

**return** *employeeCount*;

}

}

**PerformanceRating Class.**

**package** mphasis.com;

**public** **class** PerformanceRating {

**public** **static** **final** **int** ***OUTSTANDING*** = 5;

**public** **static** **final** **int** ***GOOD*** = 4;

**public** **static** **final** **int** ***AVERAGE*** = 3;

**public** **static** **final** **int** ***POOR*** = 2;

**public** **static** **int** calculatePerformance(Employees e) {

**int** points = e.getPoints();

**if** (points >= 90) {

**return** ***OUTSTANDING***;

} **else** **if** (points >= 75) {

**return** ***GOOD***;

} **else** **if** (points >= 50) {

**return** ***AVERAGE***;

} **else** {

**return** ***POOR***;

}

}

}

PerformanceCalculator Class

**package** mphasis.com;

**public** **class** PerformanceCalculato {

**public** **static** **void** main(String[] args) {

Employees emp1 = **new** Employees("Alice", 95);

Employees emp2 = **new** Employees("Bob", 82);

Employees emp3 = **new** Employees("Charlie", 67);

Employees emp4 = **new** Employees("David", 45);

System.***out***.println(emp1.getName() + "'s Performance Rating: " +

PerformanceRating.*calculatePerformance*(emp1));

System.***out***.println(emp2.getName() + "'s Performance Rating: " +

PerformanceRating.*calculatePerformance*(emp2));

System.***out***.println(emp3.getName() + "'s Performance Rating: " +

PerformanceRating.*calculatePerformance*(emp3));

System.***out***.println(emp4.getName() + "'s Performance Rating: " +

PerformanceRating.*calculatePerformance*(emp4));

System.***out***.println("Total Employees Created: " + Employees.*getEmployeeCount*());

}

}